3rd Symposium on Advances in Approximate Bayesian Inference, 2020 1–10

Distilling Ensembles Improves Uncertainty Estimates

Zelda Mariet[†], **Rodolphe Jenatton**[†], **Florian Wenzel**[⋆], **Dustin Tran**[†]
[†] *Google Brain,* [⋆] *Work done at Google Brain*

Abstract

We seek to bridge the performance gap between batch ensembles (ensembles of deep networks with shared parameters) and deep ensembles on tasks which require not only predictions, but also uncertainty estimates for these predictions. We obtain negative theoretical results on the possibility of approximating deep ensemble weights by batch ensemble weights, and so turn to distillation. Training a batch ensemble on the outputs of deep ensembles improves accuracy and uncertainty estimates, without requiring hyper-parameter tuning. This result is specific to the choice of batch ensemble architectures: distilling deep ensembles to a single network is unsuccessful, despite single networks having only marginally fewer parameters than batch ensembles.

1. Introduction and related work

With the increasingly widespread use of deep neural networks as tools to predict anything from weather patterns (Agrawal et al., 2019) to medical diagnoses (Gulshan et al., 2016), the ability to provide not only predictions but also confidence intervals for these predictions has similarly become more widely desirable.

Bayesian neural networks (Neal, 1995; MacKay et al., 1995; Blundell et al., 2015; Osawa et al., 2019; Wenzel et al., 2020a; Wilson and Izmailov, 2020), which learn a posterior distribution over the weights of the neural network, are a particularly elegant way of modeling *epistemic uncertainty*: uncertainty in the predictions that stems from the choice of model itself. By drawing multiple samples from the posterior distribution over weights, one can directly obtain a empirical estimate of the standard deviation over the resulting predictions.

However, Bayesian neural networks fall short of the mark in terms of state-of-the-art results on standard machine learning datasets, particularly in terms of accurate predictions (Ovadia et al., 2019; Wenzel et al., 2020a). For this reason, recent work on uncertainty benchmarking has focused on ensembles of deep networks ("deep ensembles"), which achieve competitive accuracy and uncertainty estimates not only on held-out test data, but also on data drawn from a distribution *shifted* away from the training distribution. This robustness makes deep ensembles a particularly attractive model choice for real-world applications that require meaningful estimates of uncertainty (Dietterich, 2000; Lakshminarayanan et al., 2017; Ovadia et al., 2019; Gustafsson et al., 2020; Wenzel et al., 2020b).

Ensembling different deep neural nets is a popular strategy to improve upon the predictive behaviors of its members (Hansen and Salamon, 1990; Dietterich, 2000). Ensembles are typically formed over the weight initialization (Lakshminarayanan et al., 2017). Duvenaud et al. (2016) showed that ensembling over random initializations can be viewed as sampling from a variational distribution fitting the Bayesian posterior.

Although deep ensembles alleviate the computational burden of training and — to some extent — inference, an ensemble of size n is also $\mathcal{O}(n)$ times larger than a single (potentially Bayesian) neural network. To reduce the memory footprint and the prediction cost of deep ensembles Wen et al. (2020) introduced *batch ensembles* (BEs): ensembles of neural networks that share a common weight parameterization, up to rank-one perturbations of their weight matrices. In addition to the memory saving, the parametrization of BEs is amenable to an efficient vectorization of the predictions, making it possible to predict with all ensemble members in a single forward pass.

In this work, we seek to bridge the performance gap between batch ensembles and deep ensembles. We first investigate the possibility of approximating the weights of a deep ensemble by the weights of a batch ensemble; however, our theoretical and empirical results show that the strong performance of batch ensembles cannot be attributed to their emulating deep ensembles. Hence, we turn to distillation to leverage the richer modeling of deep ensembles. We show that training a batch ensemble on the outputs of deep ensembles improves their accuracy and uncertainty estimates, without requiring hyper-parameter tuning. Although a batch ensemble has only marginally more parameters than a single model with the same architecture, distilled batch ensembles also significantly outperform single models trained via distillation on deep ensemble outputs.

Distillation of Bayesian networks and ensembles for better uncertainty has been explored previously, albeit not in the context of batch ensemble parameterizations. Snelson and Ghahramani (2005) and Korattikara Balan et al. (2015) distill a Monte Carlo approximation of the posterior of a Bayesian neural network into a single deterministic model. Malinin et al. (2020) distill ensembles by leveraging prior networks (Malinin and Gales, 2018), which model a conditional distribution over outputs. In (Tran et al., 2020), a teacher deep ensemble is distilled into a *multi-headed* single network.

Preliminary. We denote by $\circ$ the Hadamard product of two matrices of the same shape: $\forall i, j, (A \circ B)_{ij} = A_{ij} B_{ij}$. Similarly, we write A/B the Hadamard *quotient* of matrices A, B, as soon as B has all non-zero coefficients. We will make use of the following property of Hadamard products: for two matrices A, B of the same shape,

$$\operatorname{rank}(A \circ B) \leq \operatorname{rank}(A) \operatorname{rank}(B). \tag{1}$$

2. Are batch ensembles simply approximations of deep ensembles?

A reasonable hypothesis would explain the strong performance of batch ensembles by suggesting that, for most benchmarking tasks, weight matrices of trained deep ensembles can be approximated by a batch ensemble parameterization. In this section, we provide theoretical and empirical results showing that this explanation is unlikely.

A batch ensemble (Wen et al., 2020) imposes a shared weight structure between ensemble members. For any layer ℓ, the i-th member's weight matrix W_i from layer $\ell \to \ell+1$ satisfies

$$W_i = W_{0\ell} \circ R_i, \quad \text{with } \operatorname{rank}(R_i) = 1, \tag{2}$$

where the matrix $W_{0\ell}$ is shared across all ensemble members.

Theorem 1 *Let $W_1, \dots, W_n \in \mathbb{R}^{p \times q}$ be n weight matrices of the same shape, corresponding to n weight matrices in a deep ensemble, such that each matrix only contains non-zero coefficients. There exists a batch ensemble parameterization of $[W_i]_{i=1}^n$ given by $W_0, R_1, \dots, R_n$ such that for all i, $W_i = W_0 \circ R_i$ with $\text{rank}(R_i) = 1$ if and only if*

$$\forall 1 \leq i, j \leq n, \quad \text{rank}(W_i / W_j) = 1.$$

Proof Let us first assume there exists a batch ensemble parameterization of the weights W_i of a deep ensemble: for all i, $W_i = W_0 \circ R_i$ with $\text{rank}(R_i) = 1$. Then, for any $i \neq j \in [n]$, $W_i / W_j = R_i / R_j$. Writing $R_j = \boldsymbol{x}\boldsymbol{y}^\top$ for given vectors $\boldsymbol{x}, \boldsymbol{y}$, we have in turn $1/R_j = 1/\boldsymbol{x} \cdot (1/\boldsymbol{y})^\top$, and so $\text{rank}(1/R_j) = 1$. Hence, $\text{rank}(W_i/W_j) \leq \text{rank}(R_i)\text{rank}(1/R_j) = 1$.

Conversely, suppose that for all i, j, $\text{rank}(W_i/W_j) = 1$. Define $W_0 = W_1$ and let $R_1 = [1]_{ij}$ be the (rank-1) matrix of all-ones, and for $j \neq 1$, set $R_j = W_j / W_1$. By assumption, $\text{rank}(R_j) = 1$, and $W_0 \circ R_j = W_1 \circ (W_j / W_1) = W_i$, thereby concluding the proof. ■

Theorem 1 indicates that a set of n matrices of similar shape cannot be well-approximated by batch ensemble-style parameterization, as the space of rank-k matrices of shape $p \times q$ is not dense within the space of all matrices as soon as $k < \min(p, q)$. This result is not strictly-speaking equivalent to dismiss the possibility of parameterizing a deep ensemble as a batch ensemble: because neurons within a layer can be permuted without changing the behavior of the neural network, any weight matrix $[W_{ij}]_{i,j}$ can be represented by $[W_{\sigma(i)\tau(j)}]_{i,j}$, with σ and τ permutations of the corresponding layers. Appendix A shows how to properly generalize Theorem 1 to hold over all possible representations of a deep ensemble.

Unfortunately, initializing the weights of a batch ensemble to an approximate reconstruction[1] of the weights of a deep ensemble of the same size does not improve training. Rather, as shown in Figure 1, batch ensembles initialized with such a reconstruction are unable to learn a good model of the data, and remain well-below state-of-the-art performance on Cifar-10 (for which test accuracy lies around 96%); more results are provided in Section 4.

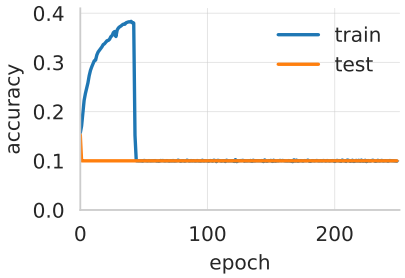

Figure 1: BE, deep ensemble init.

Remark 2 *Theorem 1 cannot be easily generalized to batch ensembles with rank $k > 1$ parameterizations, as both sides of the equivalence will have different constraints on the ranks of the Hadamard ratios, following from identity (1). However, the same identity shows that there may be hope of parameterizing dense ensembles with batch ensembles of rank $\left\lceil \sqrt{k} \right\rceil$, where k is the width of the largest layer in the neural network ensemble.*

3. Distilling deep ensembles

Given the theoretical and empirical evidence that the weights of batch-ensemble models are not simply a lighter representation of deep ensemble weights, we turn instead to learning the *function* represented by the deep model, rather than its inner parameterization. Distillation (Hinton et al., 2015) is a well-known, successful method for transferring knowledge from large models to cheaper representations.

1. We consider the matrix factorization $\min_{W_0, \{R_i\}} \sum_{i=1}^n \|W_0 \circ R_i - W_i\|^2$ solved by alternative least-squares.

We investigate the particular setting in which both teacher and student models are ensembles (Lan et al., 2018; Malinin et al., 2020; Tran et al., 2020); our hope is that the more precise modeling the teacher deep ensemble can be inherited by the student batch ensemble, despite the student's significantly reduced parameter space.

3.1. Head distillation versus knowledge distillation

When distilling from a deep ensemble of size n to a batch ensemble of equal size n, we can ask the student model to either predict the teacher prediction averaged over ensemble members, or instead match each student member to a corresponding teacher member, and do one-to-one distillation. Following the notation of (Tran et al., 2020), we refer to the former as *knowledge distillation*, and to the latter as *head distillation* (each student "head" is being matched to a teacher "head").

In head distillation, each student member emulates a specific teacher member. Intuitively, this is desirable: intra-ensemble diversity is known to be key to robustness to dataset shift and meaningful uncertainty metrics (Masegosa, 2019; Zaidi et al., 2020). With head distillation, the student model should inherit the teacher's diversity.

Conversely, knowledge distillation also presents strong methodological advantages: with the requirement of having the same ensemble size for teacher and student lifted, we can scale the size of the student model as required to increase its capacity.

4. Experimental results

We evaluate batch ensemble models distilled on the outputs of deep ensemble teachers on the Cifar-10 and Cifar-100 datasets (Krizhevsky, 2009). For both datasets, we train ensembles (resp. batch ensembles) composed of four wide-resnets (Zagoruyko and Komodakis, 2016) of shape 28-10. Results present the mean and standard deviations over 10 different initial random seeds. For all experiments, we used the hyperparameters that provide the state-of-the-art results for batch ensemble models (without distillation); notably, all our distillation results where obtained *without* any additional hyperparameter tuning.

4.1. Distillation and weight reconstruction

As baselines, we include the performance of a simple batch ensemble trained without distillation, and a single neural network ("single") distilled from the outputs of the teacher deep ensemble. Results are summarized in Tables 1 and 2; DE Teacher records the performance of the (deep ensemble) teacher model, which is fixed across all experiments.

We also include results when initializing with the weights of the deep ensemble. As discussed above, however, batch ensembles are rarely able to recover from the initialization, and their performance is poor. We do not include the results for this method on Cifar-100, as it did not perform meaningfully better than random.

Distilling the deep ensemble architecture into a student batch ensemble drastically improves the calibration, on standard and corrupted test data (Ovadia et al., 2019). For uncertainty metrics — ECE (Expected Callibration Error) and NLL (Negative Log Likelihood) — this allows the batch ensemble to match the performance of the teacher deep ensemble, despite a 4× decrease in memory footprint. Crucially, this improvement in cali-

Table 1: Cifar-10 results. The teacher model is a deep ensemble of 4 wide-resnets 28-10. All batch ensembles are of size 4 and provide a factorized form of the wide-resnet 28-10. Distilled batch ensembles match or slightly improve upon the accuracy of standard batch ensembles, and significantly improves performance on calibration metrics.

Model	Distill	Head	Acc. ↑	ECE ↓	NLL ↓	c-Acc. ↑	c-ECE ↓	c-NLL ↓
DE Teacher			96.8	0.009	0.155	0.784	0.086	1.015
BE			**0.962** ± 0.001	0.018 ± 0.001	0.139 ± 0.005	**0.782** ± 0.006	**0.120** ± 0.004	0.950 ± 0.031
Lone Student	✓		0.960 ± 0.001	0.022 ± 0.001	0.153 ± 0.005	0.762 ± 0.003	0.150 ± 0.004	1.019 ± 0.027
BE + Init	×		0.448 ± 0.367	0.031 ± 0.033	0.602 ± 0.046	0.324 ± 0.236	**0.049**±0.052	1.423 ± 0.025
BE + Init	✓	×	0.315 ± 0.337	0.049 ± 0.092	0.844 ± 0.422	0.234 ± 0.212	**0.039**±0.062	1.508 ± 0.228
BE Student	✓	✓	**0.963**±0.001	**0.016**±0.001	**0.132**±0.003	**0.779**±0.008	0.119 ± 0.005	**0.911**±0.044
BE Student	✓	×	0.962 ± 0.001	0.017 ± 0.001	**0.132**±0.002	**0.778**±0.004	0.119 ± 0.003	**0.903**±0.019

bration is not accompanied by a degradation of accuracy metrics. We also report accuracy, ECE and NLL on corrupted data (c-Acc., c-ECE, c-NLL), as in (Ovadia et al., 2019).

In contrast, distilling a deep ensemble into a single wide-resnet performs significantly worse than distilling to a batch ensemble, despite the small difference in number of parameters (36.5M for a single wide-resnet, 36.6M for a batch ensemble of size 4).

Surprisingly, head distillation does not provide meaningful improvements over knowledge distillation; this result is in line with those presented in (Tran et al., 2020). We speculate that this is due to the same reason that underlies the poor performance of batch ensembles initialized with deep ensemble weights. Namely, constraining batch ensembles to emulate the inner representation of deep ensembles — either by weight initialization or matching per-member predictions — is harmful.

Table 2: Cifar-100 results. All models have the same architecture as in Table 1. Once again, distilled batch ensembles match or slightly improve upon the accuracy of standard batch ensembles, and significantly improves performance on calibration metrics.

Model	Distill	Head	Acc. ↑	ECE ↓	NLL ↓	c-Acc. ↑	c-ECE ↓	c-NLL ↓
DE Teacher			0.819	0.021	0.833	0.526	0.135	2.758
BE			**0.818** ± 0.003	0.025 ± 0.004	0.696 ± 0.012	**0.531** ± 0.004	0.145 ± 0.010	2.565 ± 0.055
Lone student	✓		0.798 ± 0.003	0.073 ± 0.003	0.829 ± 0.006	0.513 ± 0.003	0.221 ± 0.009	2.597 ± 0.067
BE Student	✓	✓	**0.819**±0.003	**0.018**±0.002	**0.676**±0.009	**0.535**±0.004	**0.128**±0.004	**2.393**±0.031
BE Student	✓	×	**0.820**±0.003	**0.019**±0.005	**0.672**±0.007	**0.534**±0.004	**0.130**±0.010	**2.397**±0.045

4.2. Varying student ensemble size

As head distillation does not improve upon knowledge distillation results, it is natural to take advantage of the comparative lack of constraints of knowledge distillation to learn batch ensembles of increasing size for fixed teacher ensemble sizes. Figures 2 and 3 show the impact of the student size on the different evaluation metrics.

On the Cifar-10 dataset, increasing the ensemble size immediately improve performance across all accuracy and uncertainty metrics, on both standard and corrupted test data. On Cifar-100, the best batch ensemble size appears to be 4, after which performance on the standard Cifar-100 test set saturates, then begins to decay. However, on corrupted data, increasing the batch ensemble size continues to yield meaningful improvements for uncertainty estimates, while decreasing accuracy no more than by one percentage point.

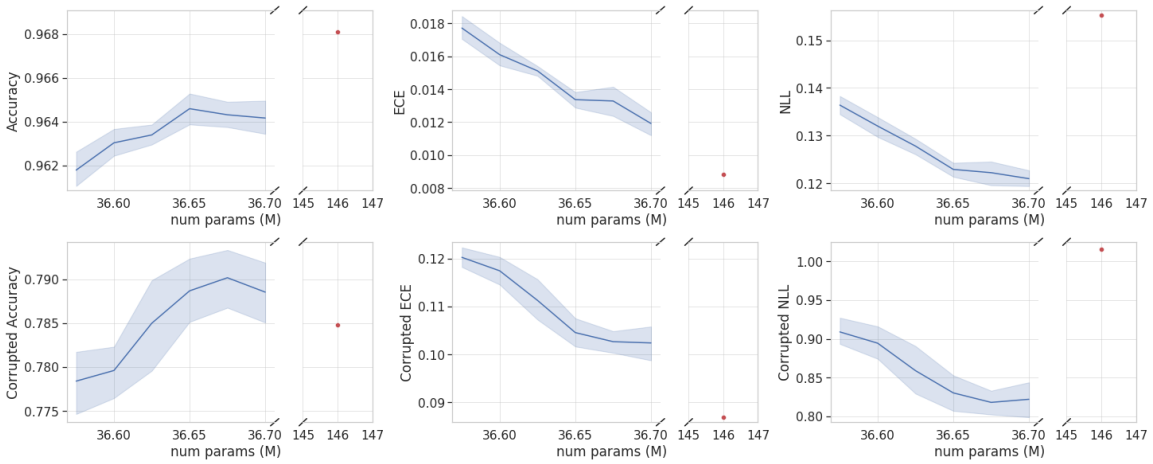

Figure 2: Performance on Cifar10 when increasing the number of members from 3 to 8 of a batch ensemble; the red circle marker indicates the performance of a deep ensemble of size 4.

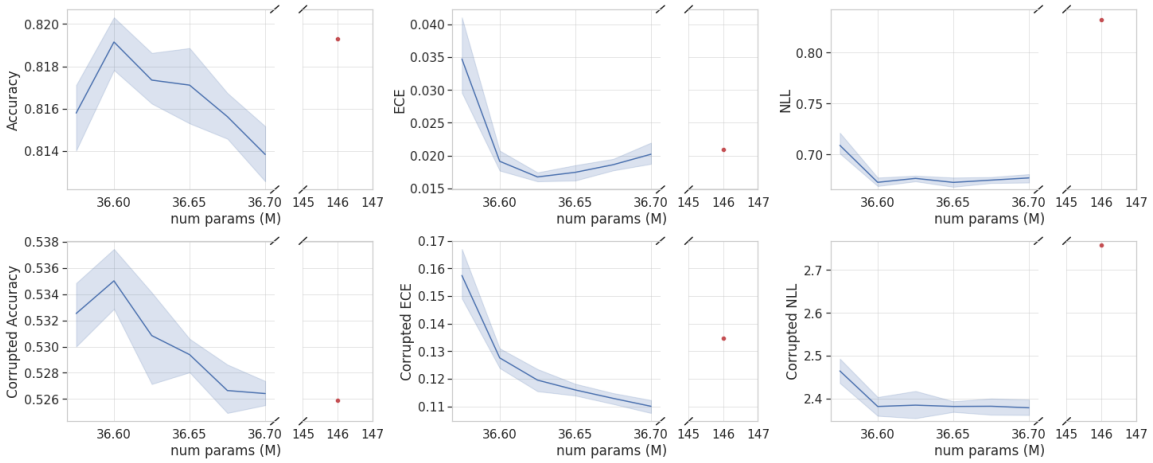

Figure 3: Performance on Cifar100 when increasing the number of members from 3 to 8 of a batch ensemble; the red circle marker indicates the performance of a deep ensemble of size 4.

5. Conclusion and future work

Knowledge distillation is an inexpensive way of narrowing the gap between batch and deep ensembles. As knowledge distillation only relies on teacher predictions, regardless of teacher structure, knowledge distillation provides a generic and modular framework within which architectures of teacher and student can be changed independently, e.g., using recent work such as (Wenzel et al., 2020b; Zaidi et al., 2020). However, preliminary results (App. B.1) suggest that hyperparameter diversity cannot be naively inherited by batch ensembles.

Surprisingly, an analysis of the disagreement between ensemble members did not show improved diversity in distilled ensembles; similarly, the variance term introduced in (Masegosa, 2019) did not increase with distillation. This suggests that the improvements are not due to the student models inheriting the diversity from their teachers, and leaves open the question of the underlying cause of success for distillation.

References

Shreya Agrawal, Luke Barrington, Carla Bromberg, John Burge, Cenk Gazen, and Jason Hickey. Machine learning for precipitation nowcasting from radar images. *arXiv preprint arXiv:1912.12132*, 2019.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, 2015.

Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

David Duvenaud, Dougal Maclaurin, and Ryan Adams. Early stopping as nonparametric variational inference. volume 51 of *Proceedings of Machine Learning Research*, Cadiz, Spain, 09–11 May 2016. PMLR.

Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22), 2016.

Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020.

Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10), 1990.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

Anoop Korattikara Balan, Vivek Rathod, Kevin P Murphy, and Max Welling. Bayesian dark knowledge. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Xu Lan, Xiatian Zhu, and Shaogang Gong. Knowledge distillation by on-the-fly native ensemble. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018.

David JC MacKay et al. Ensemble learning and evidence maximization. In *Advances in neural information processing systems*, 1995.

Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble distribution distillation. In *International Conference on Learning Representations*, 2020.

Andrés R. Masegosa. Learning from i.i.d. data under model miss-specification. *CoRR*, abs/1912.08335, 2019.

Radford M Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.

Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz E Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. Practical deep learning with bayesian principles. In *Advances in neural information processing systems*, 2019.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019.

Edward Snelson and Zoubin Ghahramani. Compact approximations to bayesian predictive distributions. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, New York, NY, USA, 2005. Association for Computing Machinery. doi: 10.1145/1102351.1102457.

Linh Tran, Bastiaan S. Veeling, Kevin Roth, Jakub Swiatkowski, Joshua V. Dillon, Jasper Snoek, Stephan Mandt, Tim Salimans, Sebastian Nowozin, and Rodolphe Jenatton. Hydra: Preserving ensemble diversity for model distillation, 2020.

Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In *International Conference on Learning Representations*, 2020.

Florian Wenzel, Kevin Roth, Bastiaan S Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? In *International Conference on Machine Learning*, 2020a.

Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. Hyperparameter ensembles for robustness and uncertainty quantification. In *Advances in Neural Information Processing Systems*, 2020b.

Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In *International Conference on Machine Learning*, 2020.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, September 2016. doi: 10.5244/C.30.87.

Sheheryar Zaidi, Arber Zela, Thomas Elsken, Chris Holmes, Frank Hutter, and Yee Whye Teh. Neural ensemble search for performant and calibrated predictions. *arXiv preprint arXiv:2006.08573*, 2020.

Appendix A. Theoretical results

Remark 3 *Theorem 1 must be generalized to the case where for each i, there exists two permutation matrices P, Q of appropriate size such that $\frac{PW_iQ}{W_1}$ is of rank 1. This provides a more natural characterization of weights in neural networks, since the input-output neurons may need to be reordered to compare weight matrices across ensemble members.*

As there exists a finite number of permutation matrices of a given shape, the theoretical results from Theorem 1 can be generalized to this setting easily.

Appendix B. Additional experimental results

B.1. Distilling from hyper-deep ensembles

We additionally investigated the possibility of distilling a batch ensemble from a *hyper* deep ensemble (Wenzel et al., 2020b); results are reported in Tables 3 and 4. Note that the teacher performance reported is still the performance of the deep ensemble teacher.

Unfortunately, batch ensembles are not able to benefit from the additional lift provided by hyper deep ensembles.

Table 3: Cifar-10 results, including results when distilling from a hyper deep ensemble of size 4.

Model	Distill	Head	Acc. ↑	ECE ↓	NLL ↓	c-Acc. ↑	c-ECE ↓	c-NLL ↓
DE Teacher			96.8	0.009	0.155	0.784	0.086	1.015
BE			**0.962** ± 0.001	0.018 ± 0.001	0.139 ± 0.005	**0.782** ± 0.006	**0.120** ± 0.004	0.950 ± 0.031
Lone Student	✓		0.960 ± 0.001	0.022 ± 0.001	0.153 ± 0.005	0.762 ± 0.003	0.150 ± 0.004	1.019 ± 0.027
BE Student	✓	✓	**0.963**±0.001	**0.016**±0.001	**0.132**±0.003	**0.779**±0.008	**0.119**±0.005	**0.911**±0.044
BE Student	✓	×	0.962 ± 0.001	**0.017**±0.001	**0.132**±0.002	**0.778**±0.004	**0.119**±0.003	**0.903**±0.019
Hyper Student	✓	✓	0.962 ± 0.001	**0.017**±0.001	**0.132**±0.003	**0.779**±0.003	**0.118**±0.004	**0.898**±0.020
Hyper Student	✓	×	**0.963**±0.002	**0.016**±0.001	**0.130**±0.004	**0.777**±0.006	**0.119**±0.006	**0.916**±0.045

Table 4: Cifar-100 results, including results when distilling from a hyper deep ensemble of size 4.

Model	Distill	Head	Acc. ↑	ECE ↓	NLL ↓	c-Acc. ↑	c-ECE ↓	c-NLL ↓
DE Teacher			0.819	0.021	0.833	0.526	0.135	2.758
BE			**0.818** ± 0.003	0.025 ± 0.004	0.696 ± 0.012	**0.531** ± 0.004	0.145 ± 0.010	2.565 ± 0.055
Lone Student	✓		0.798 +- 0.003	0.073 +- 0.003	0.829 +- 0.006	0.513 +- 0.003	0.221 +- 0.009	2.597 +- 0.067
BE Student	✓	✓	**0.819**+-0.003	**0.018**+-0.002	**0.676**+-0.009	**0.535**+-0.004	**0.128**+-0.004	**2.393**+-0.031
BE Student	✓	×	**0.820**+-0.003	**0.019**+-0.005	**0.672**+-0.007	**0.534**+-0.004	**0.130**+-0.010	**2.397**+-0.045
Hyper Student	✓	✓	**0.819**+-0.003	0.022 +- 0.001	0.682 +- 0.009	**0.534**+-0.004	0.134 +- 0.003	2.441 +- 0.041
Hyper Student	✓	×	**0.820**+-0.001	0.022 +- 0.002	0.679 +- 0.006	**0.535**+-0.004	**0.131**+-0.005	**2.422**+-0.046

B.2. Regularizing batch ensemble weights to deep ensemble weights

In a second attempt to recover deep ensemble weights within batch ensembles, we impose a L_2 regularization term with varying levels of strength during training on the difference between the deep ensemble weights and the batch ensemble weights. However, as shown in Table 5, this does not improve upon distillation results. Results obtained when simply imposing the regularization without distillation are not included, as they were not as strong as when using distillation.

Table 5: Cifar-100 results. All models have the same architecture as in Table 1. Once again, distilled batch ensembles match or slightly improve upon the accuracy of standard batch ensembles, and significantly improves performance on calibration metrics.

Model	Distill	Head	Acc. ↑	ECE ↓	NLL ↓	c-Acc. ↑	c-ECE ↓	c-NLL ↓
DE Teacher			0.819	0.021	0.833	0.526	0.135	2.758
BE			**0.818** ± 0.003	0.025 ± 0.004	0.696 ± 0.012	**0.531** ± 0.004	0.145 ± 0.010	2.565 ± 0.055
Lone Student	✓		0.798 +- 0.003	0.073 +- 0.003	0.829 +- 0.006	0.513 +- 0.003	0.221 +- 0.009	2.597 +- 0.067
BE Student	✓	✓	**0.819**+-0.003	**0.018**+-0.002	**0.676**+-0.009	**0.535**+-0.004	**0.128**+-0.004	**2.393**+-0.031
BE Student	✓	×	**0.820**+-0.003	**0.019**+-0.005	**0.672**+-0.007	**0.534**+-0.004	**0.130**+-0.010	**2.397**+-0.045
Reg. (0.01)	✓	✓	**0.821**+-0.003	0.020 +- 0.004	**0.674**+-0.006	**0.533**+-0.006	0.132 +- 0.009	**2.406**+-0.059
Reg. (0.01)	✓	×	**0.820**+-0.001	0.021 +- 0.002	**0.675**+-0.005	**0.534**+-0.007	**0.129**+-0.011	**2.390**+-0.078
Reg. (0.1)	✓	✓	**0.821**+-0.002	**0.019**+-0.001	**0.672**+-0.004	**0.537**+-0.004	**0.129**+-0.006	**2.361**+-0.052
Reg. (0.1)	✓	×	**0.820**+-0.002	**0.019**+-0.002	**0.672**+-0.007	**0.535**+-0.004	**0.127**+-0.006	**2.378**+-0.041
Reg. (1.0)	✓	✓	**0.820**+-0.002	**0.019**+-0.002	**0.675**+-0.009	**0.534**+-0.003	0.132 +- 0.009	**2.409**+-0.057
Reg. (1.0)	✓	×	**0.820**+-0.003	**0.019**+-0.002	**0.674**+-0.007	**0.536**+-0.004	**0.126**+-0.004	**2.378**+-0.038